

Claim Amendments

1. (Currently Amended) A compiling method, comprising
compiling source code into source binary code for a first computing platform;
and

generating an annotation section associated with the source binary code, the
annotation section comprising an annotation for a scope, the scope comprising at
least one block of the source binary code having at least one attribute to aid a
translator optimization,

wherein the at least one attribute comprises at least one of information
associated with register spilling and restoring instructions within the at least one block,
information associated with local variable assignment within the at least one block and
information associated with volatile variable access within the at least one block.

2. (Original) The compiling method of claim 1, wherein if the scope comprises a
plurality of blocks, the blocks have consecutive addresses with each other and have
the at least one attribute in common.

3. (Original) The compiling method of claim 1, wherein the annotation section
further comprises a region annotation for a region comprising one or more scope.

4. (Original) The compiling method of claim 3, wherein the annotation for the scope further comprises scope addresses, scope size and the at least one attribute, and the annotation for the region further comprises a region annotation pointer, region addresses and region size.

5-7. (Canceled)

8. (Currently Amended) A translating method, comprising:

inputting source binary code for a first computing platform and an annotation section associated with the source binary code; and

translating the source binary code to a target binary code for a second computing platform by utilizing the annotation section,

wherein, an annotation section comprises an annotation for a scope, the scope comprising at least one block of the source binary code having at least one attribute to aid a translator optimization, and wherein the at least one attribute comprises at least one of information associated with register spilling and restoring instructions within the at least one block, information associated with local variable assignment within the at least one block and information associated with volatile variable access within the at least one block.

9. (Original) The translating method of claim 8, wherein if the scope comprises a plurality of blocks, the blocks have consecutive addresses with each other and have the at least one attribute in common.

10. (Original) The translating method of claim 8, wherein the annotation section further comprises an annotation for a region comprising one or more scope.

11. (Original) The translating method of claim 10, wherein the annotation for the scope further comprises scope addresses, scope size and the at least one attribute, and the annotation for the region further comprises a region annotation pointer, region addresses and region size.

12-14. (Canceled).

15. (Original) The translating method of claim 8, wherein translating the source binary code for the source platform further comprises:

generating target intermediate code based upon the source binary code;

optimizing the target intermediate code by utilizing the annotation section; and

generating the target binary code for the target platform based upon the optimized target intermediate code.

16. (Original) The translating method of claim 15, wherein optimizing the target intermediate code further comprises:

generating an internal representation for the annotation section in response to determining that the internal representation has not been established;

reading from the annotation section an attribute associated with a block of the target intermediate code based upon the internal representation; and

optimizing the block based upon the read attribute.

17. (Original) The translating method of claim 16, wherein the internal representation is an AVL tree, a node of the AVL tree comprising region addresses and region annotation pointer.

18. (Currently Amended) A compiler, comprising:

a compiling component to compile source code into source binary code for a first computing platform;

an annotation generator to generate an annotation section associated with the source binary code, the annotation section comprising an annotation for a scope, the scope comprising at least one block of the source binary code having at least one attribute to aid a translator optimization,

wherein the at least one attribute comprises at least one of information associated with register spilling and restoring instructions within the at least one block,

information associated with local variable assignment within the at least one block and
information associated with volatile variable access within the at least one block.

19. (Original) The compiler of claim 18, wherein if the scope comprises a plurality of blocks, the blocks have consecutive addresses with each other and have the at least one attribute in common.

20. (Original) The compiler of claim 18, wherein the annotation section further comprises an annotation for a region comprising one or more scope.

21. (Original) The compiler of claim 20, wherein the annotation for the scope further comprises scope addresses, scope size and the at least one attribute, and the annotation for the region further comprises a region annotation pointer, region addresses and region size.

22-24. (Canceled).

25. (Currently Amended) A translator, comprising:

an input component to input source binary code for a first computing platform and an annotation section associated with the source binary code; and

a translate component to translate the source binary code to target binary code for a second computing platform by utilizing the annotation section,

wherein, the annotation section comprises an annotation for a scope, the scope comprising at least one block of the source binary code having at least one attribute to aid a translator optimization, wherein the at least one attribute comprises at least one of information associated with register spilling and restoring instructions within the at least one block, information associated with local variable assignment within the at least one block and information associated with volatile variable access within the at least one block.

26. (Original) The translator of claim 25, wherein if the scope comprises a plurality of blocks, the blocks have consecutive addresses with each other and have the at least one attribute in common.

27. (Original) The translator of claim 26, wherein the annotation section further comprises an annotation for a region comprising one or more scope.

28. (Original) The translator of claim 27, wherein the annotation for the scope further comprises scope addresses, scope size and the at least one attribute, and the annotation for the region further comprises a region annotation pointer, region addresses and region size.

29. (Original) The translator of claim 25, wherein the translate component generates target intermediate code based upon the source binary code; optimizes the target intermediate code by utilizing the annotation section; and generates the target binary code for the target platform based upon the optimized target intermediate code.

30. (Original) The translator of claim 25, wherein the translate component generates an internal representation for the annotation section in response to determining that the internal representation has not been established; read from the annotation section an attribute associated with a block of the target intermediate code based upon the internal representation; and optimizes the block based upon the read attribute.